PROCESSING

VLSI Coprocessor Delivers High Quality Displays

Many microprocessor-based systems today use VLSI technology in processing and memory components. However, designers of subsystems have, up until now, not been able to incorporate this technology into their products because of the lack of available ICs. When, in 1981, NEC introduced the 7220 graphics display controller, users found that they could bolster system performance by off-loading graphics control chores from the system CPU. Second-sourced by Intel as the 82720, the chip uses its own drawing processor to access the required positions in the bitmap and handles both processing and display functions.

Now, Intel is poised to introduce a text coprocessor, the 82730, which is specifically tailored to execute data manipulation and display tasks. Lucio Lanza of Intel explains, "In an intelligent terminal or workstation, the CPU spends a lot of its time manipulating both graphics and text. We have identified these areas in terms of CPU use and we have distributed these blocks so that the CPU is not overburdened."

Coprocessors fall into two categories based on their architecture and operation. One type expands the microprocessor's own architecture by adding additional hardware and instructions. This type of tightly coupled coprocessor can be thought of as a transparent expansion of the microprocessor's architecture and works in sychronization with the CPU. Intel's first such coprocessor, the 8087, was designed for numerics processing and increased the microprocessor's math performance as much as 100 times.

The second type of coprocessor independently fetches its own data and sends instructions in parallel to the microprocessor. It therefore allows the microprocessor to process the tasks it handles best and delegate to the coprocessor the task it is best equipped to handle. In this cate-

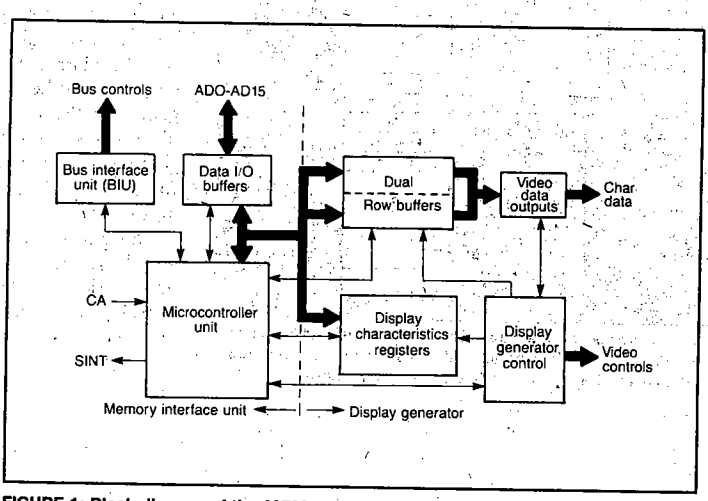

FIGURE 1: Block diagram of the 82730.

Andrew Wilson
Technical Editor

Reprinted from ELECTRONIC IMAGING © April 1983, Morgan-Grampian Publishing Company, Boston, MA 02215
50 Order Number: 231307-001

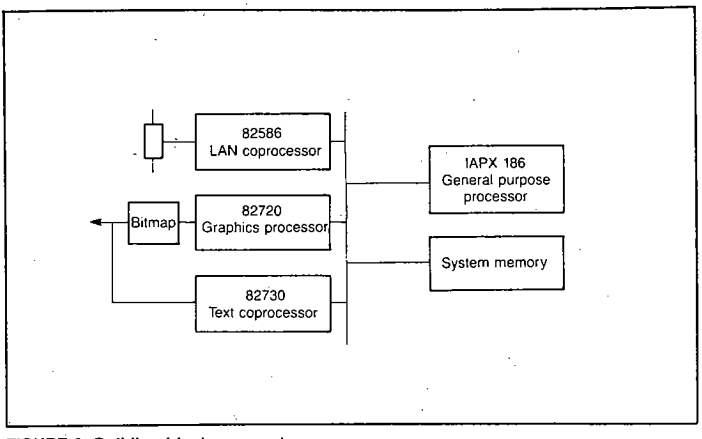

FIGURE 2: Building block approach.

gory are I/O channel coprocessors and others that deal with communications and text processing tasks.

"The 82720 is not yet at this level," Lanza said, "since it does not have the capability of going to memory and extracting its own instruction and executing it—it needs something to spoon feed it."

Coprocessors of the second category do not monitor the CPU instruction stream. Instead, they are linked to the CPU via messages prepared and stored in shared memory. The CPU will prepare data and high level directives and then place them in memory. Upon completion of this control block, the CPU will alert the coprocessor by signaling it through a common channel attention line. From that point on, the coprocessor works on its own, fetching required data and instructions and then executing those instructions.

It is not synchronized with the CPU but works asynchronously and independently. When the coprocessor completes its task, it informs the CPU by signaling on the CPU's interrupt line.

The rationale for designing a coprocessor with one or the other architectures depends on the application requirement. Tightly coupling the coprocessor with the CPU gives the advantage of a short coprocessor preparation time but has the drawback of consuming the CPU's bus bandwidth.

In the case of numeric processing, the speed of executing the floating point algorithm is of paramount importance. Reducing the preparation time of the coprocessor task is the key because of the number of microseconds it takes to execute the task. Rapid algorithmic execution requires tight coupling. In the application of the I/O related coprocessor, the task execution time is much longer and the requirement for bus time can be much higher. And, for I/O operations the preparation time is not critical. A shared memory coupling is preferred for those types of applications because it provides greater flexibility in the design of the bus structure.

Text coprocessing

"In the design of the 82730," said Lanza, "we have tried to eliminate all the known differences between what is visible on the screen and what is obtained on the printed page. In word processing systems today, even the length of a row on the CRT is sometimes not the same as the length seen in print. Clearly, when you are editing text this becomes a major problem."

The 82730 supports the generation of text displays through features which include proportional spacing, simultaneous superscript/subscript, dynamically reloadable fonts and user programmable field and character attributes. Editing capabilities are further enhanced by features such as split screen, virtual windows, smooth scrolling and table-driven linked lists.

Figure 1 shows a block diagram of the 82730. The chip is divided into two main sections—the memory interface unit and the display generator. The memory interface unit provides the communication between the 82730 and the system processor, while the display generator acts on the display data and carries out the display operation.

Communication between the 82730 and the CPU takes place through messages placed in communication blocks in shared memory. The processor issues channel commands by preparing these message blocks and directing the 82730's attention to them by activating a hardware channel attention signal (CA). The memory interface unit fetches and executes these commands. When the display process is activated, the 82730 repeatedly fetches display data and embedded datastream commands from memory utilizing its built-in DMA capability, executes any datastream commands as encountered on the fly, and loads the row buffers with the display data. After executing these commands, the 82730 clears a busy flag in memory, to inform the host CPU that it is ready for the next command.

The memory interface unit is divided into two sections—the bus unit and the microcontroller unit. The bus interface unit provides the electrical interface to the system bus and the timing signals required for the microcontroller unit operations, making these operations transparent to the microcontroller unit. The 82730 can be programmed during initialization to provide 8 or 16 bit data, and 16 or 32 bit addressing.

The microcontroller unit contains the microinstruction store and the associated circuitry required for the execution of all channel and datastream commands. It uses the bus interface unit in carrying out its memory access tasks such as loading the row buffers with display data.

The interaction between the microcontroller unit and the display generator takes place through shared internal storage. The microcon-

"The device provides the ability to independently maximize the performance of the CPU."

troller unit fetches data from memory and writes it in the internal storage, while the display generator reads from the internal storage and carries out the display operation. The microcontroller unit and display generator operate asynchronously with respect to each other. Synchronization is accomplished through communication via internal flags and display timing signals generated by the display generator. The internal shared storage consists of row buffers which store the display data and internal registers which store display parameters. There are two row buffers each capable of storing up to 200 characters. The data in one row buffer is used by the display generator to display one complete character row on the screen, while the microcontoller unit is loading the second row buffer with display data fetched from memory. At the end of the row being displayed, the buffers are swapped and the microcontroller unit and display generator resume their tasks.

The display characteristics registers contain all the information used to control every aspect of display characteristics from screen size to blink rates. A major portion of this register set is the three content addressable memory (CAM) arrays that allow flexible timing control for row and screen characteristics. The user has the power to set the parameters for the entire screen by invoking a single high-level command.

By separating the video interface clocks from the bus interface clock, the 82730 provides the designer with the ability to independently maximize the performance of the CPU and video sections of the system.

The video interface consists of two independent clocks: the Reference Clock (RCLK) and the Character Clock (CCLK). While the RCLK controls the raster timing and defines the screen layout, the CCLK independently shifts character and attribute information out of the 82730, which allows proportional spacing to be achieved.

Combining text and graphics

A major requirement in the design of engineering workstations is the simultaneous display of both text and graphics. In terms of graphics requirements, the designer of such systems needs a drawing processor for fast geometric primitives, a math processor for fast transformations and a general purpose processor for access to the graphics database.

For text, string processing is needed for manipulation of text primitives and database processing is needed for access to the document files. The solution to this problem can be solved by using both the 720 graphics coprocessor and the 730 text coprocessor (Figure 2).

Both coprocessors work with Intel's new 82586 communications coprocessor. This works in conjunction with a CPU and the appropriate software to provide local area network (LAN) control capabilities. Message data to be placed on the network by a microprocessor-based work station is stored in shared memory in transmit blocks. Pointers (starting address information) to these blocks are stored along with processing instructions in other shared memory blocks. Status information and overall directives are stored in system control blocks which serve as the mailbox between the CPU and the 82586.

When alerted by a channel attention signal, the 82586 will perform a host of tasks involved in accessing data to be transmitted from its location in memory, framing the message packets containing the data and seeing to the transmission on the network medium. In addition, the 82586 receives and buffers incoming data which it then stores in shared memory for the CPU to collect. It is the CPU's job to allocate the blocks of memory for the LAN coprocessor to store the received packet data.